ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

RASPBERRY PET PAL: FINAL REPORT FOR ECE445

Team #31

YIROU JIN
(yirouj2@illinois.edu)
SHUHAN GUO
(shuhan4@illinois.edu)
XIAOMIN QIU
(xchou2@illinois.edu)
XIAOSHAN WU
(xw50@illinois.edu)

Sponsor: Gaoang Wang, Said Mikki
TA: Yiqun Niu

May 23, 2023

Abstract

Loneliness and financial constraints have led many young people to seek companionship in pets. However, owning a real pet can be costly and challenging. To address this, we have designed EmoEmo, an affordable and functional electronic pet robot that offers voice and behavior interaction. EmoEmo's functionalities are controlled through voice commands, and it can provide engaging responses and perform various tasks. The design incorporates open-source wake word technology to awaken the pet and Baidu Brain's voice control technology to convert user commands into text. EmoEmo can interact with ChatGPT to generate AI responses and utilize edge-tts speech synthesis technology for speech output. The robot is equipped with immersive and non-immersive plugins, allowing users to engage with specific functionalities while also using general commands. Immersive plugins enable continuous interaction, while non-immersive plugins perform one-time operations. EmoEmo's capabilities include following objects, displaying emotions on its screen, recognizing and responding to voice commands, avoiding obstacles, answering questions, and more. With cutting-edge technology, advanced features, and a user-friendly interface, EmoEmo provides an interactive and enjoyable experience for users of all ages.

Contents

1 Introduction **1**
- 1.1 Problem and Solution Overview . . . 1
- 1.2 High-Level Requirements List . . . 2

2 Design **3**
- 2.1 System Overview . . . 3
  - 2.1.1 Block Diagram . . . 3
- 2.2 Physical Structure . . . 4
- 2.3 Motion Unit . . . 7
  - 2.3.1 Motor Module . . . 7
  - 2.3.2 Sensor Module . . . 8
  - 2.3.3 Obstacle Avoidance Algorithm . . . 8
- 2.4 Face Tracking Unit . . . 9
  - 2.4.1 Pan-Tilt Hat . . . 9
  - 2.4.2 Face Recognition Algorithm . . . 10
- 2.5 Voice Control Unit . . . 13
  - 2.5.1 Hotword Spotting . . . 13
  - 2.5.2 Voice Recognition . . . 13
  - 2.5.3 Plugins . . . 13
  - 2.5.4 Voice Synthesis . . . 14
  - 2.5.5 OLED Screen . . . 14
- 2.6 Tolerance Analysis . . . 15

3 Verification **18**
- 3.1 Motion Unit . . . 18
- 3.2 Voice Control Unit . . . 18
- 3.3 Face Tracking Unit . . . 19

4 Cost **20**

5 Conclusion **21**
- 5.1 Accomplishments . . . 21
- 5.2 Uncertainties . . . 21
- 5.3 Ethics . . . 22
- 5.4 Future Work . . . 23

References **25**

1 Introduction

1.1 Problem and Solution Overview

In November 2022, OpenAI released ChatGPT, a powerful tool for interactive dialogue. While ChatGPT has opened up new possibilities, there is a need for further exploration in various fields, including robotics and the Internet of Things. Integrating ChatGPT with devices such as robots and smart homes has the potential to enhance their responsiveness and flexibility. This graduation project is based on the concept of integrating large language models (LLMs) with robots, aiming to leverage the capabilities of ChatGPT to create an intelligent electronic pet robot, EmoEmo.

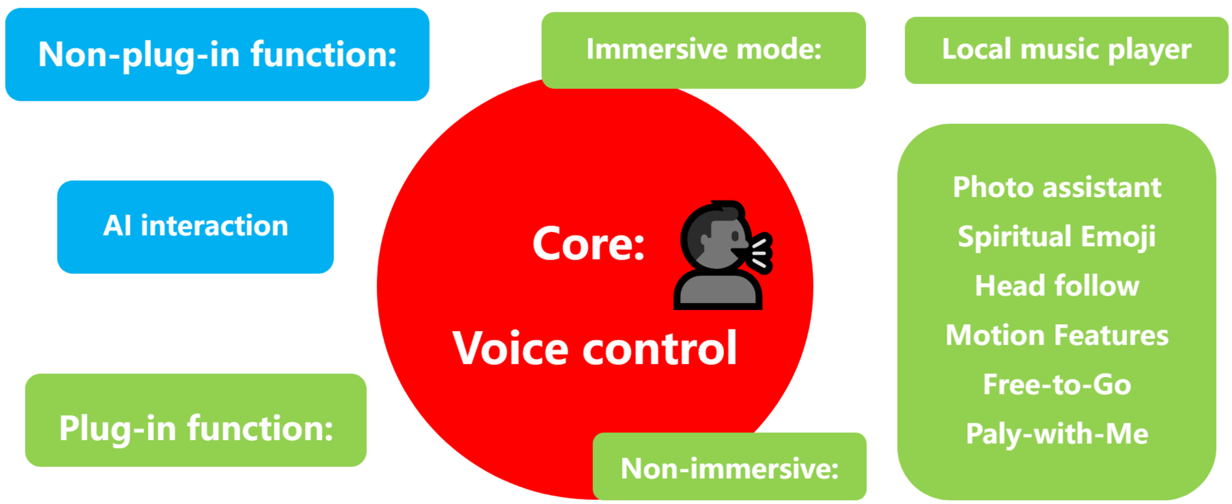

Figure 1: The overall functions display. The whole system is controlled by user's voice, which there are three specific kinds of functions, i.e., AI conversation, immersive plugin and non-immersive plugins.

In this graduation project, we have designed an intelligent electronic pet robot, EmoEmo, which is centered around voice control. As shown in 1, initially, the user must awaken the pet by saying "EmoEmo". To achieve this functionality, we have utilized the open-source wake word project, Snowboy. Upon hearing the ding sound indicating that it has been awakened, EmoEmo receives user commands through voice, which are then converted into text using Baidu Brain's voice control technology. After receiving the command, EmoEmo first determines whether the command contains the keywords required to activate any custom plugins. Plugins refer to the custom functions we have designed for EmoEmo. If a plugin is not required, EmoEmo will forward the command to the ChatGPT interface, receive the AI's response, and then convert it into speech using edge-tts speech synthesis technology, which will be played back to the user via a Bluetooth speaker. If a plugin is required, EmoEmo will begin to execute the corresponding functionality. Plugins are divided into immersive and non-immersive plugins. Non-immersive plugins are one-time operations that cannot be interrupted during execution. After completing the operation, EmoEmo will automatically enter a waiting state until it is reawakened by the user. The majority of our plugins are non-immersive. Immersive plugins, on the other hand, allow users to continue to awaken the electronic pet and issue commands while the plugin is running. Immersive plugins contain some commands that are only valid during their activation. Of course, while using immersive plugins, users can not only use plugin-specific commands but also use the aforementioned general commands. If a user wishes to stop an immersive plugin, they must use a specific termination command.

The electronic pet is able to perform all the desired functions reliably and accurately. It should be able to follow objects, display a range of emotions on its screen, recognize and respond to voice commands, avoid obstacles, and answer the owner's questions. It will be equipped with cutting-edge technology and advanced features for an interactive and engaging experience. It can follow the ball's movements through target detection technology while displaying a range of expressions through a high-quality display screen. With its voice recognition and corresponding audio output, this pet can communicate with its owner and respond to commands. Additionally, it can track the owner's face through its facial detection capabilities, and navigate its surroundings using obstacle avoidance technology. Furthermore, the pet can take photos and play music for the owner. With its advanced features, this electronic pet offers unparalleled interactivity and companionship for pet lovers of all ages.

1.2 High-Level Requirements List

- Functionality: The electronic pet should be able to perform all the desired functions reliably and accurately. It should be able to follow objects, display a range of emotions on its screen, recognize and respond to voice commands, avoid obstacles, and interact with pan-tilt using face following.

- User experience: The electronic pet should be easy to use and interact with. Users should be able to easily control and communicate with the pet through its display screen and voice recognition system. The pet should also respond to user interactions fun and engagingly.

- Durability and stability: The electronic pet should be built using durable and stable components to ensure that it can withstand regular use without breaking down. The car should be stable enough to navigate different terrains and avoid obstacles without getting stuck or tipping over.

2 Design

2.1 System Overview

2.1.1 Block Diagram

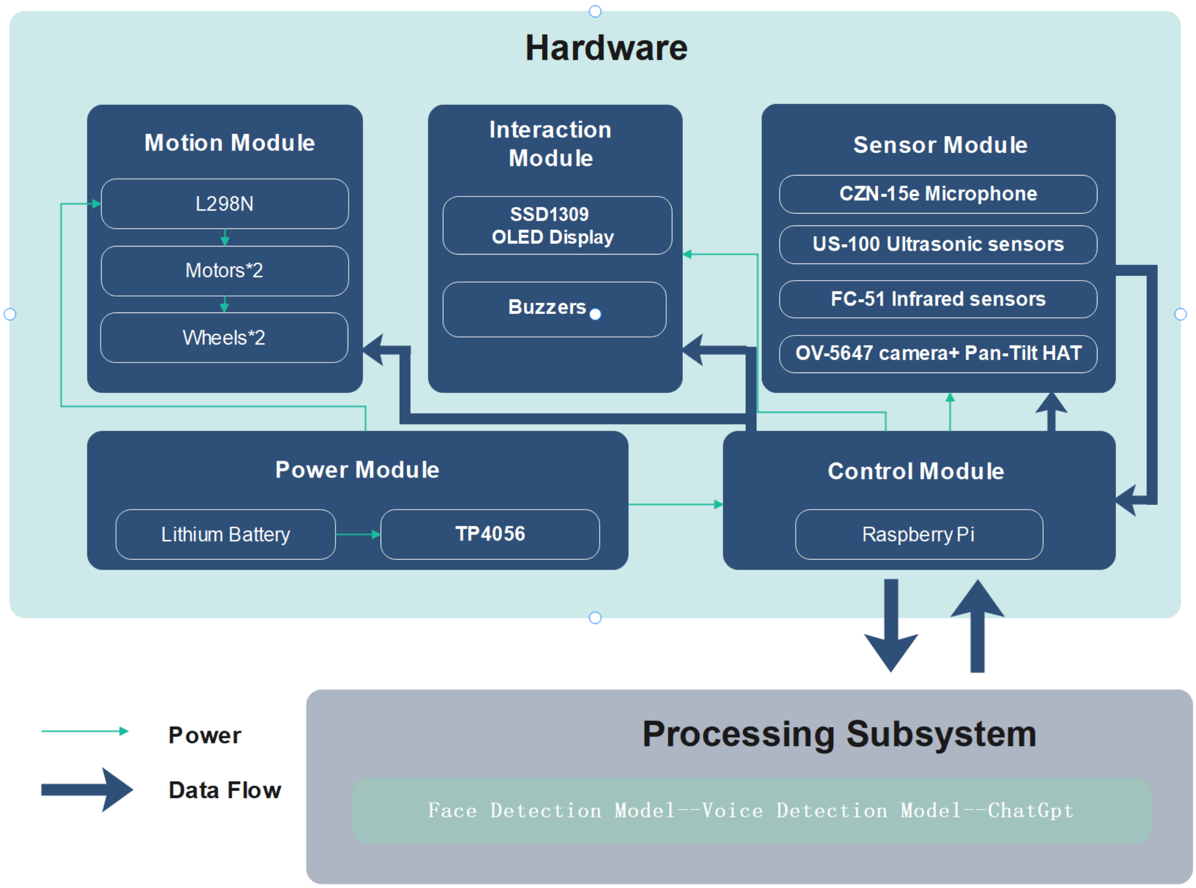

Figure 2: Our system consists of software and hardware components. In terms of hardware, we have five modules: control, motion, interaction, sensor, and power. The control module is implemented by Raspberry Pi, which serves as the hub connecting our software and hardware. The software controls the hardware through the Raspberry Pi. The power module relies on lithium batteries for power supply and uses a protection circuit to prevent damage to the Raspberry Pi and sensors. The motion module relies on L298N to control two motors to achieve pet movement. The interaction module includes voice interaction and display interaction. The sensor module captures external information through ultrasonic sensors, infrared sensors, cameras, microphones, etc.

2.2 Physical Structure

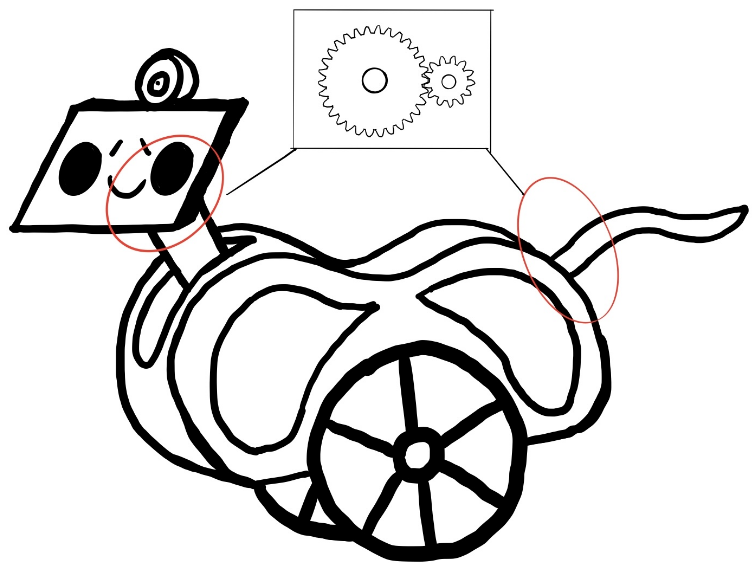

Figure 3: Original Design Sketch

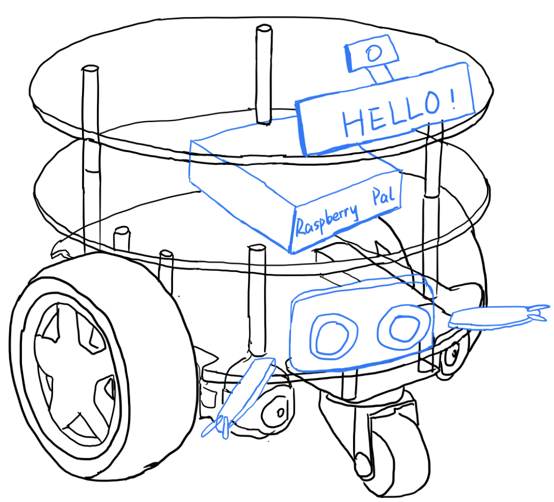

Figure 4: Recent Design Sketch

In sketch, the physical structure is a three-layer car. The base layer is the bottommost layer of the car, which acts as the foundation for the other layers. It is usually a flat and rigid board that holds the motors, wheels, battery, and other components like ultrasonic sensors, and infrared sensors. The base layer may also have mounting holes for attaching the other layers. The control layer is the middle layer of the car, which houses the Raspberry Pi board and other control electronics. The control layer is responsible for processing the sensor data and sending commands to the motors. The interactive layer is the topmost layer of the car, which is designed to hold the pan-tilt camera and LED screen. The whole appearance will be designed to be very cute.

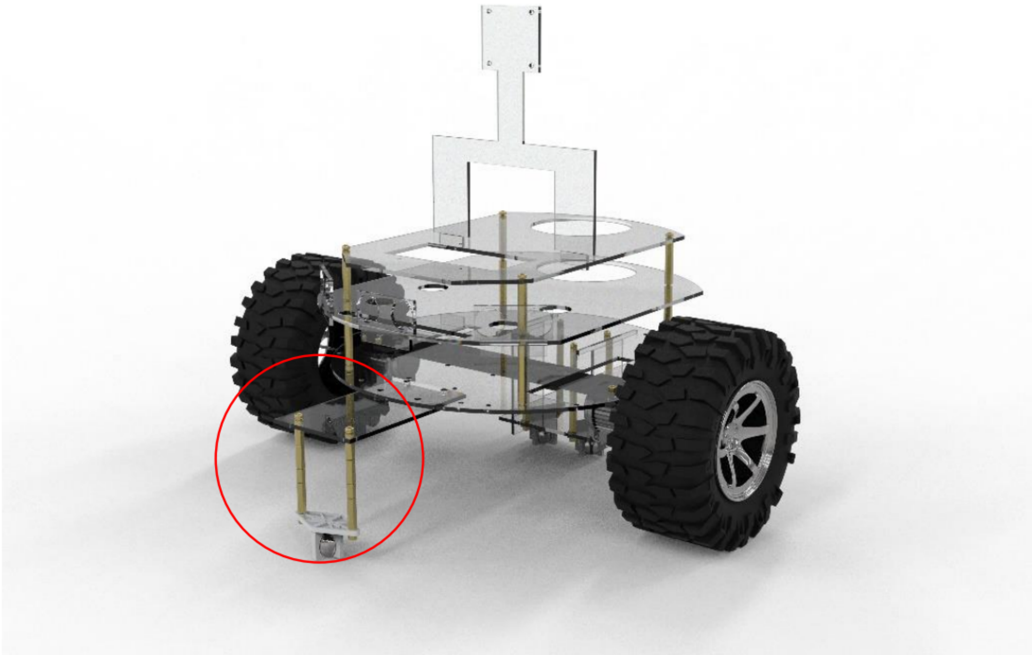

Figure 5: CAD model view 1

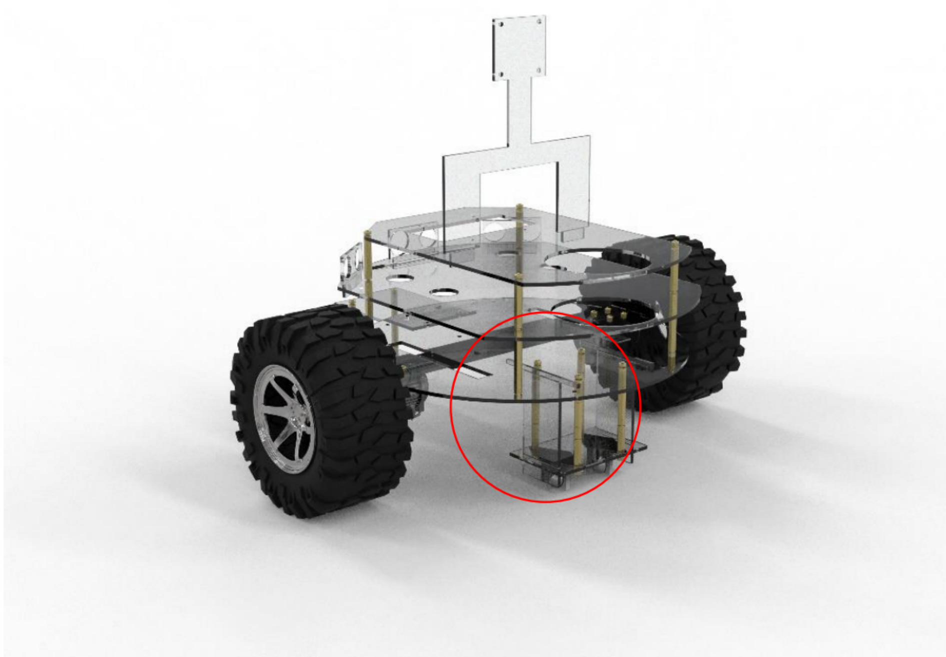

Figure 6: CAD model view 2

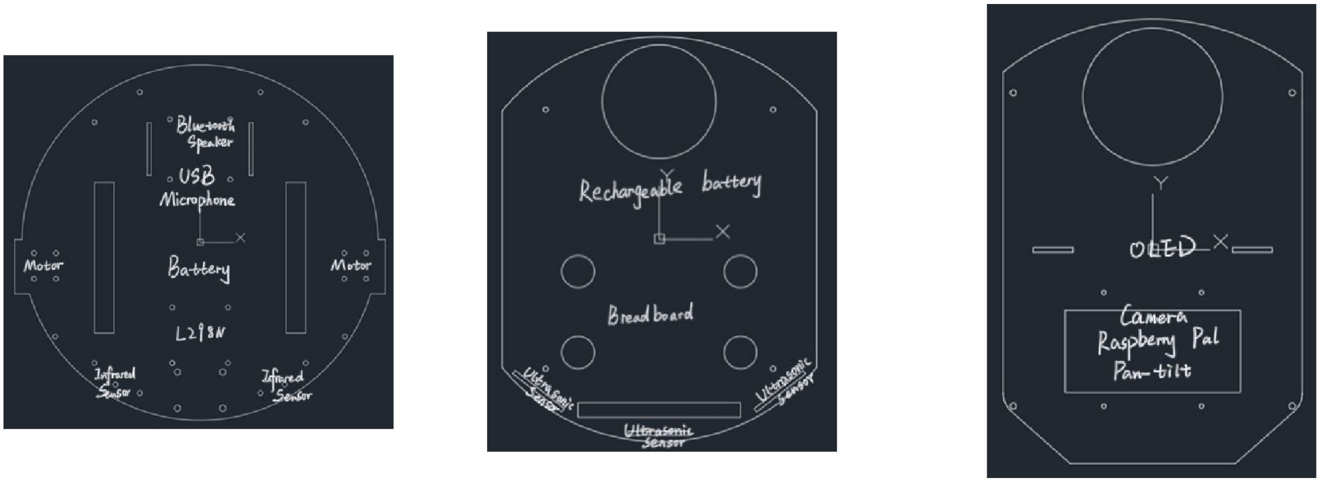

Figure 7: Components Position

The CAD model design is shown above. On the bottom plate, the motor, L298N, a pack of two lithium batteries, two infrared sensors and USB microphone are placed. And on the medium plate, three ultrasonic sensors, breadboard and rechargeable battery are placed, while Raspberry Pi, Pan-tilt Hat, CSI camera, OLED Screen are placed on the top plate. Note that I drew the red circle in 5, this extension out of the place can prevent the car suddenly braked and rolled forward. Moreover, note that the red circle in 6, this space can be used to place the Bluetooth speaker, not only the sound will not be blocked by the shell, and to balance the weight, there is a ground support force to provide a smoother movement of the car.

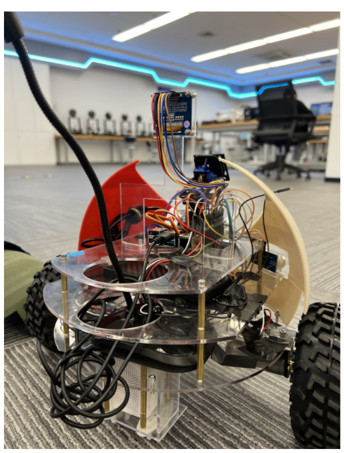

Figure 8: Final inner physical structure

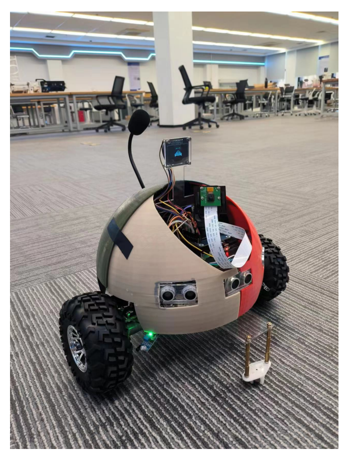

Figure 9: Final Physical Structure

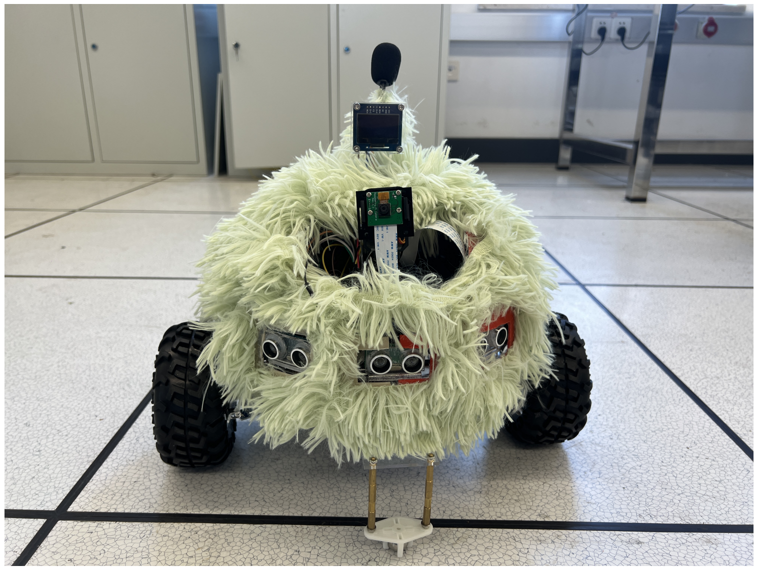

Figure 10: Final Physical Structure with fur

2.3 Motion Unit

2.3.1 Motor Module

The **motion module** consists of the motor driver board, motors, and wheels. The motor module is connected to the Raspberry Pi through GPIO pins, and the Python-written method is used to control the motor speed and achieve differential drive control of the two wheels. The primary function of this module is to move the pet robot.

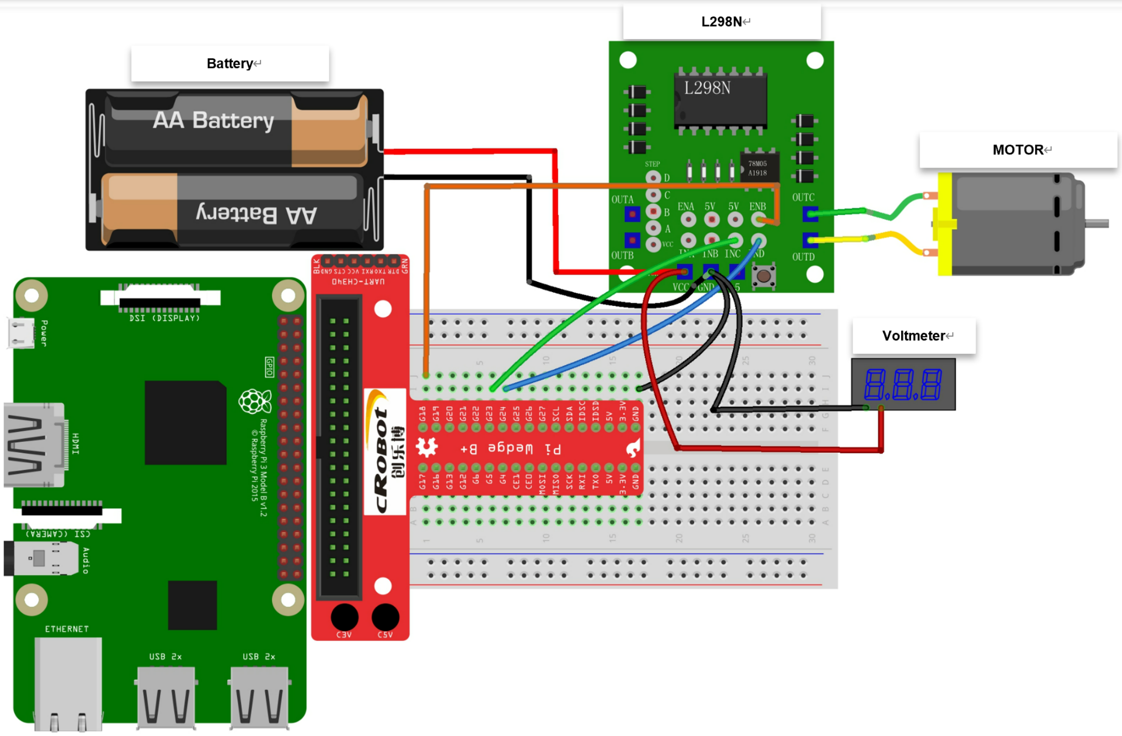

Figure 11: Motion Module

Controlling the steering and speed of DC motors is actually achieved by setting different values for the control class pins.

2.3.2 Sensor Module

1) Ultrasound Sensor: In our design, three ultrasonic sensors are used to detect the distance between the cart and the left, middle and right obstacles.

2) Infrared Sensor: The principle of infrared sensor is that, under the power-on state, the infrared transmitter head emits infrared signal, and after the target reflection the receiver head receives this signal and outputs a low-level signal, the Raspberry Pi collects this low-level signal and takes corresponding measures to avoid obstacles. The infrared sensor has a sensitivity adjustment option, the sensor detection distance can be adjusted by potentiometer, clockwise potentiometer, the detection distance increases: counterclockwise potentiometer, the detection distance decreases.

2.3.3 Obstacle Avoidance Algorithm

The Obstacle Avoidance Algorithm uses the data collected by the sensor module to make decisions. The FC-51 infrared sensor returns left_measure and right_measure to show whether there are obstacles in the left or right. And the ultrasonic sensor returns the distance data and the control module would use this to set speed by pwm.

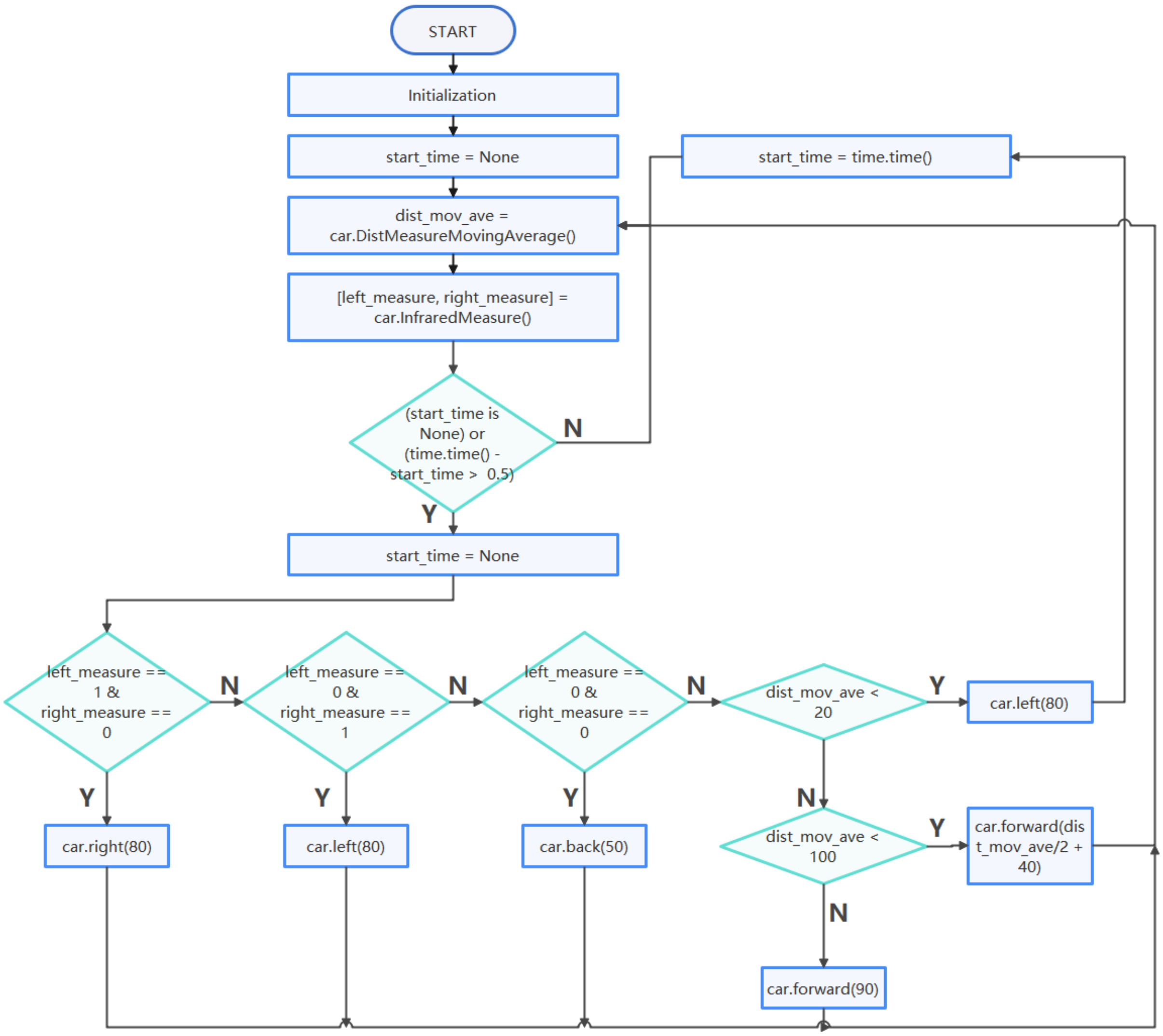

Figure 12: Obstacle Avoidance Flow Chart

2.4 Face Tracking Unit

2.4.1 Pan-Tilt Hat

The controller of this product is PCA9685, a 12-bit precision 16-channel PWM wave output chip based on IIC bus communication. And the on-board TSL2581 ambient light sensor, by detecting the light intensity to assist the camera work, also through the I2C interface control, will not occupy too much interface pin resources.

The Pan-tilt Hat uses I2C communication, one data line and one clock line. The I2C bus has three types of signals during data transfer: start signal, end signal and answer signal. To connect the Pan-tilt Hat to Raspberry Pi, we need to enable the I2C interface first, and then install the BCM2835 library and the WiringPi library, after that we can control the horizontal and tilt movement of the Pan-tilt Hat through python code.

2.4.2 Face Recognition Algorithm

The purpose of the face tracking algorithm is to detect and track human faces in images or videos. It is commonly used in various applications such as computer vision, image processing, surveillance systems, augmented reality, and human-computer interaction.

The face tracking algorithm typically involves multiple steps, including face detection, face recognition, and face tracking. The algorithm analyzes the input data (images or video frames) to identify the presence and location of faces. Once a face is detected, the algorithm may extract facial features, such as eyes, nose, and mouth, to identify specific facial landmarks or to estimate head pose and facial expressions.

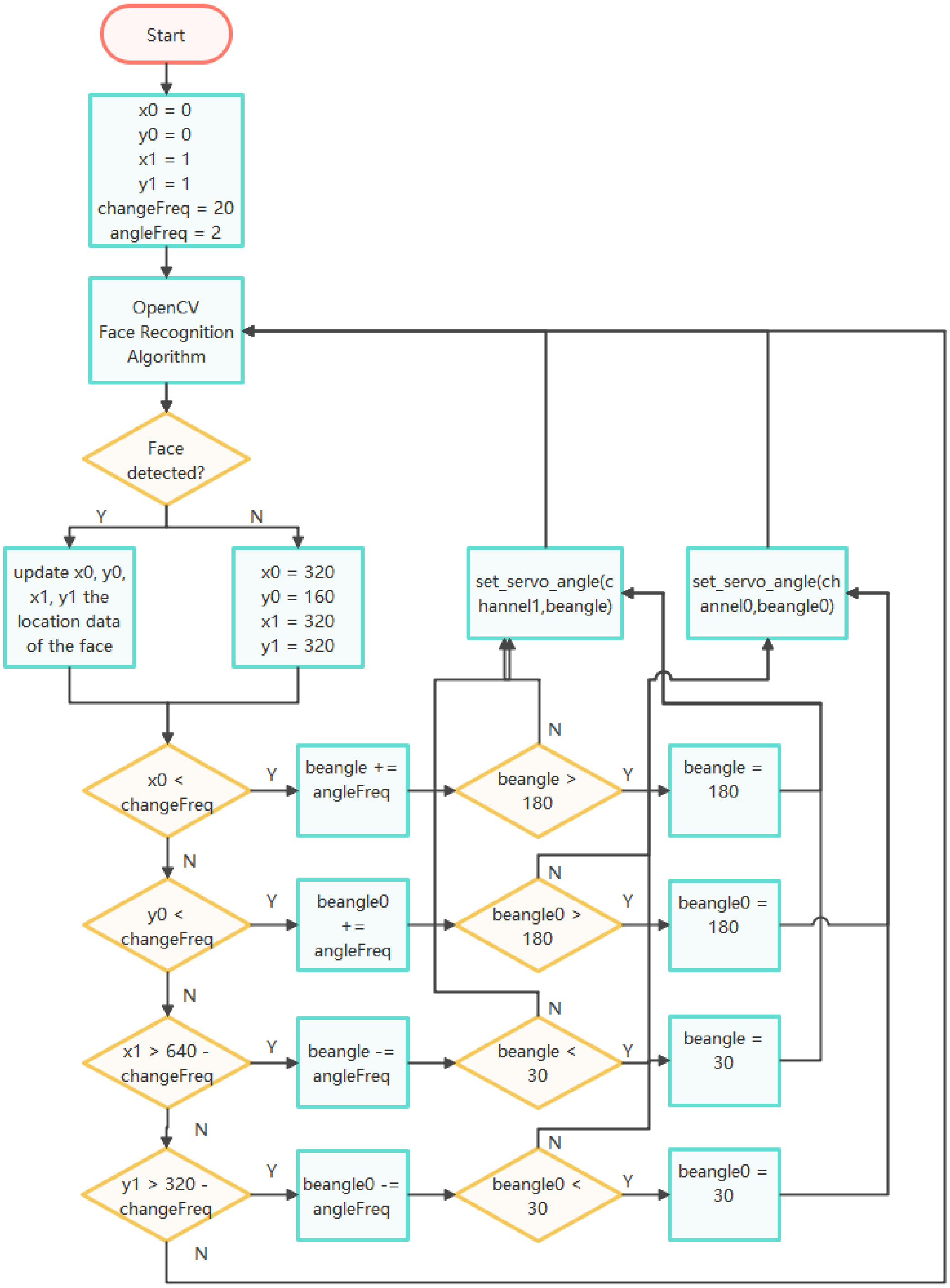

Figure 13: Face Tracking Algorithm

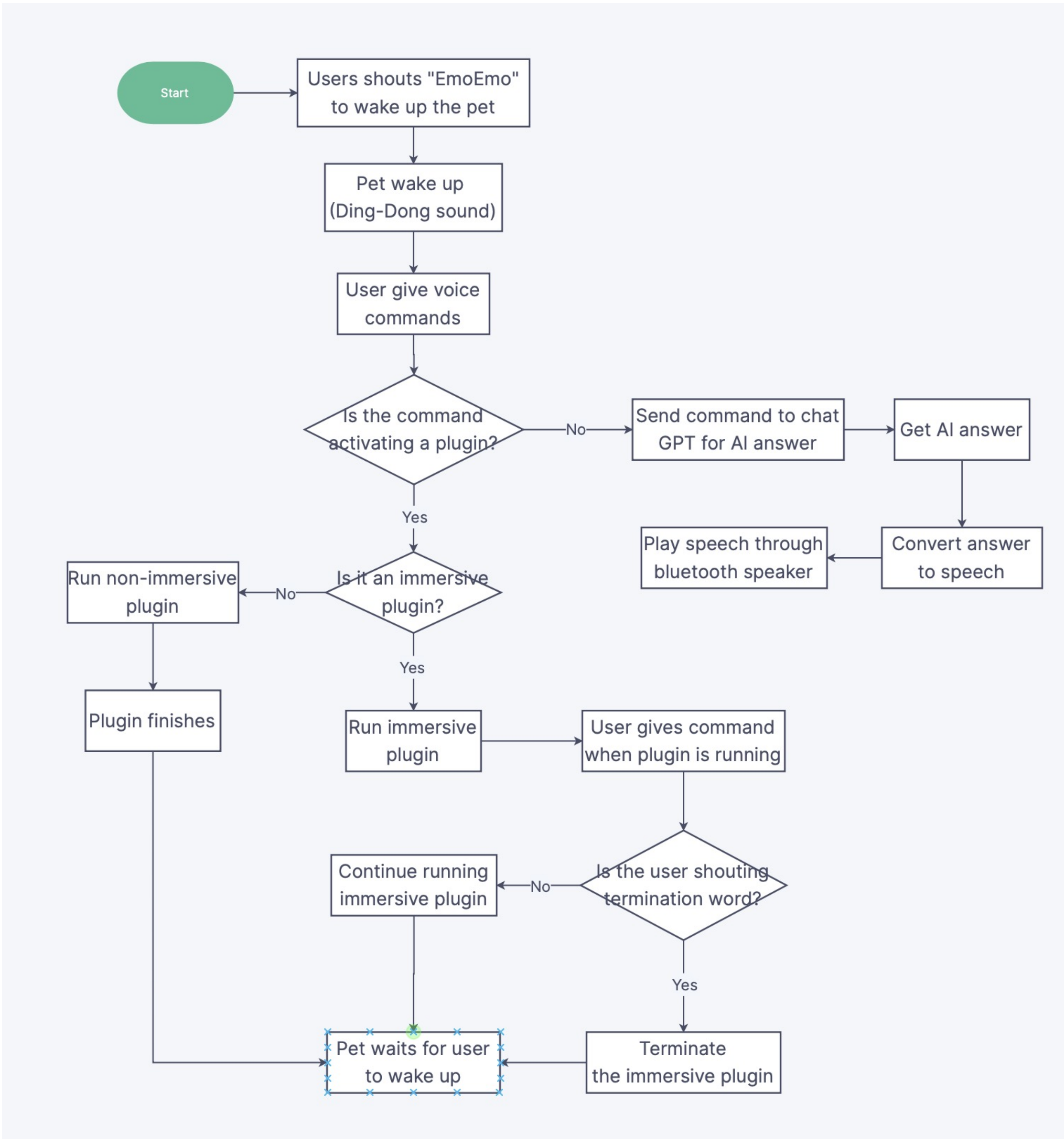

Figure 14: The working flow of EmoEmo. As shown in the graph, the whole system is voice-controlled. All the functions begin with a keyword spotting, in our case, it is EmoEmo. After waking the pet pal up with EmoEmo, the user can say any order to it. It will first check if this order can activate a plugin. If it can, the system will run the corresponding functions. If not, it will ask GPT-3.5 for the answer. The plugins here are Musics, Movements, and Camera.

2.5 Voice Control Unit

As introduced in Section Introduction, the whole working process is voice controlled, and the working flow is shown in Figure 14. This function is designed based on wukong-robot [1].

2.5.1 Hotword Spotting

Since the whole process is controlled by voice, the E-pet needs a mechanism to tell valid order from random speaking noise. Therefore, we introduced the hotword spotting mechanism, that is, only the order following the hotword is valid, just like ”Hi Siri”. As shown in Figure 14, the implementation of all features is contingent upon the initial activation of the electronic pet through the hotword wake-up. In our case, the word is ”EmoEmo”. The open-source project snowboy [2] is used to implement the hotword wake-up mechanism. During operation, EmoEmo captures audio input from the microphone during the listening phase and matches and recognizes the input using an acoustic model. If the audio input matches the acoustic features of the wake-up word, EmoEmo triggers the corresponding operation or callback function. Specifically, the hotword wake-up function of EmoEmo is primarily implemented through the *HotwordDetector* function. This function opens the recording device and continuously stores the recorded sound in a buffer. A thread within the function continuously loops to detect the hotword. The *silent_count_threshold* parameter defines the length of the silence that signals the end of a phrase during recording, while the *recording_timeout* parameter defines the maximum duration of a recording. These two parameters work together to determine when the next round of detection will begin.

2.5.2 Voice Recognition

We utilize the HTTP interface of the Baidu Speech Recognition engine [3] to achieve speech recognition. Specifically, we first convert the WAV recording files, previously collected between the start and end prompt tones after the E-pet is woken up by EmoEmo via a USB microphone, to PCM format audio data. Next, we use the Baidu Speech Recognition API to recognize the speech contained in the PCM data, returning the recognized result.

2.5.3 Plugins

Plugins refer to user-defined functions that can be activated by specific keywords. Upon receiving the text output from voice recognition, the system checks if any plugin-related keywords are present.

If no keywords are detected, the system forwards the text to the OpenAI API for further processing.

If a keyword is detected, the corresponding plugin function is executed, enabling the user to perform customized actions based on their specific needs.

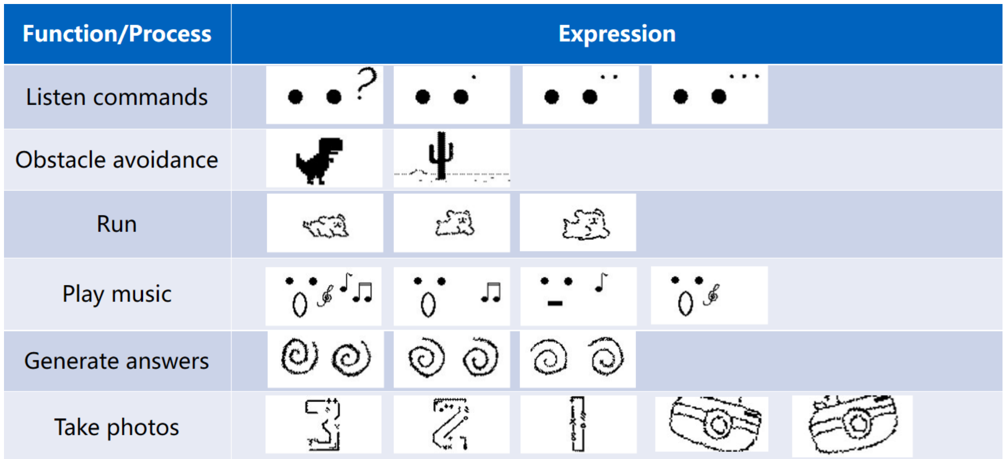

Figure 15: Some example images shown on the OLED screen.

2.5.4 Voice Synthesis

This electronic pet utilizes the text-to-speech (TTS) technology of Edge TTS [4] for speech synthesis. The process involves searching for the corresponding speech in the speech library based on the returned words, setting different speech intervals according to punctuation marks, and ultimately synthesizing the speech and intervals into a coherent voice segment.

Specifically, when a word is returned, the device searches for the corresponding speech segment in the speech library. The device then sets different speech intervals based on punctuation marks in the text, such as comma, period, or exclamation mark. Finally, the synthesized speech segments and intervals are combined to form a cohesive and natural-sounding voice segment.

The Edge TTS technology used in this electronic pet is a highly effective and efficient method for achieving speech synthesis on edge devices. This technology enables the device to perform speech synthesis tasks in real-time without the need for cloud connectivity, making it ideal for applications where internet connectivity is limited or not available.

2.5.5 OLED Screen

The OLED thread operates independently from the voice interaction thread and interacts with it through a global variable. The different stages of the voice interaction thread modify this global variable, which the OLED thread uses to adjust the displayed content.

Specifically, the OLED thread functions as a separate thread of execution, operating independently from the voice interaction thread. To communicate with the voice interaction thread, the OLED thread uses a global variable. The voice interaction thread modifies this global variable during different stages of the interaction, and the OLED thread updates the displayed content accordingly.

This approach allows for a seamless and natural user experience by ensuring that the OLED display responds in real-time to the user's voice interactions. By utilizing a global variable for communication between the threads, this approach minimizes delays and disruptions in the user experience, resulting in a more efficient and effective system overall.

2.6 Tolerance Analysis

Motor driver board compatibility issues: If the motor driver board is not compatible with the motors or the Raspberry Pi, it may not function properly or could potentially damage the components. This could be mitigated by carefully selecting a motor driver board that is rated for the appropriate voltage and current and has the necessary features to support the required number of motors. This interface is responsible for translating the commands received by the Raspberry Pi into the necessary signals to drive the motors. Any variation or error in this interface can lead to erratic behavior or complete failure of the car.

To analyze this interface mathematically, we can consider the specifications of the motor controller board and the requirements of the Raspberry Pi. Let us assume that the motor controller board can receive commands from the Raspberry Pi with a voltage range of 3.3V to 5V, and can drive the motors with a current range of 0A to 5A. The Raspberry Pi sends commands to the motor controller board through a GPIO (General Purpose Input/Output) pin.

To ensure that the Raspberry Pi car operates correctly, we need to ensure that the voltage and current levels at the interface between the Raspberry Pi and the motor controller board are within acceptable tolerances. Let us assume that the allowable voltage tolerance is +/- 0.1V and the allowable current tolerance is +/- 0.1A.

If we consider a worst-case scenario where the Raspberry Pi sends a command with a voltage of 3.3V and the motor controller board outputs a current of 5A, we can calculate the maximum voltage drop and current rise that would still be within tolerances. The maximum allowable voltage drop would be 0.1V, which means that the voltage at the interface could drop to 3.2V and still be within tolerance. The maximum allowable current rise would be 0.1A, which means that the current at the interface could rise to 5.1A and still be within tolerance.

To implement this interface with the required tolerances, we can use a voltage regulator and a current limiter circuit at the output of the Raspberry Pi's GPIO pin. The voltage regulator would ensure that the voltage at the interface is always within the allowable tolerance range, while the current limiter would ensure that the current at the interface is always within the allowable tolerance range. These circuits can be designed and implemented using off-the-shelf components such as voltage regulators and current limiters, and can be tested and validated to ensure that they meet the required specifications.

The weight of our Raspberry pet pal is 2.36kg. The coefficient of friction on tile floor with stripes is at least 0.5. The diameter of the tire is 130mm. The maximum torque of the

motor makes 9kg. Thus, the torque is enough to drive the whole car.
The torque each tire needed to overcome friction:
τ = F $\cdot r = \frac{\mu F_N}{2} r = \frac{0.5\times2.36\times9.81}{2} \times \frac{130\times10^{-3}}{2} = 0.38\,\text{Nm}$

Power supply issues: If the lithium battery voltage drops too low or the Raspberry Pi and motor driver board are not properly protected, it could result in damage to the components or loss of power. This could be mitigated by selecting a lithium battery with the appropriate voltage and capacity for the project, using a charging and protection circuit to ensure safe operation, and monitoring the battery levels to prevent over-discharging. To analyze the tolerance requirements of the power supply, we can use statistical methods such as Monte Carlo simulation to determine the effects of manufacturing variations on the power supply voltage. We can model the power supply voltage as a normal distribution with a mean voltage value and a standard deviation that represents the manufacturing tolerance.

Assume that the required voltage for the Raspberry Pi car is 5V, and the manufacturing tolerance for the power supply is +/- 0.1V. We can model the power supply voltage as a normal distribution with a mean value of 5V and a standard deviation of 0.1V. Using Monte Carlo simulation, we can generate a large number of random samples from this distribution and calculate the percentage of samples that fall within the acceptable voltage range.

Suppose we require the power supply voltage to be within +/- 0.2V of the target voltage. In that case, we can calculate the percentage of samples that fall within this range using the cumulative distribution function of the normal distribution. If this percentage is high enough to meet our requirements, then we can conclude that the power supply tolerance is feasible and can be implemented.

Suppose the calculated percentage is not high enough to meet our requirements. In that case, we may need to adjust the power supply design or specifications to increase the tolerance or reduce the manufacturing variations to ensure that the power supply can provide a stable and consistent voltage to the Raspberry Pi car.

To verify the motion module, I performed tests to measure the speed and torque of the motor at different PWM duty cycles. I used a digital multimeter and an oscilloscope to measure the voltage and current across the motor and verified that the readings were within expected ranges. We used Raspberry pi as a power source to supply L298N, when motor was running, the voltages of L298N's outputs is fluctuating, ranging from 3V to 5V, which can't ensure proper operation of motors. Thus, we deiced to use another power system a pack of two lithium batteries to power L298N separately.

AI model size: Considering the computational power of raspberry Pi, we should choose the lite weight AI model for inference. we can use the following formula to estimate the memory required for a neural network:

Memory (MB) = (Number of Parameters x Precision) / (8 x 1024 x 1024)

Where:

- Number of Parameters: The number of parameters in the neural network
- Precision: The precision of the data (e.g. 32-bit, 16-bit, etc.)
- 8 x 1024 x 1024: Conversion factor to convert bytes to megabytes

Assuming a neural network with 100 million parameters and a precision of 16 bits, the memory required would be:

Memory (MB) = (100,000,000 x 16) / (8 x 1024 x 1024) = 190.73 MB

This calculation shows that a neural network with 100 million parameters and a precision of 16 bits would require approximately 190 MB of memory. This can be a challenging requirement for the Raspberry Pi Car, which has limited memory resources.

Physical Structure Simulation The physical structure of the car is likely to be its stability and balance, as this affects its ability to navigate and maneuver. To analyze the tolerance of the physical structure of the Raspberry Pi car, we need to consider the critical dimensions that affect its stability and balance. These dimensions might include the distance between the wheels, the height of the car, the weight distribution of the components, and the alignment of the motors and sensors. Here, we do force simulation on the bottommost layer, while the weight above is 2720g and the material of layer is Acrylic.

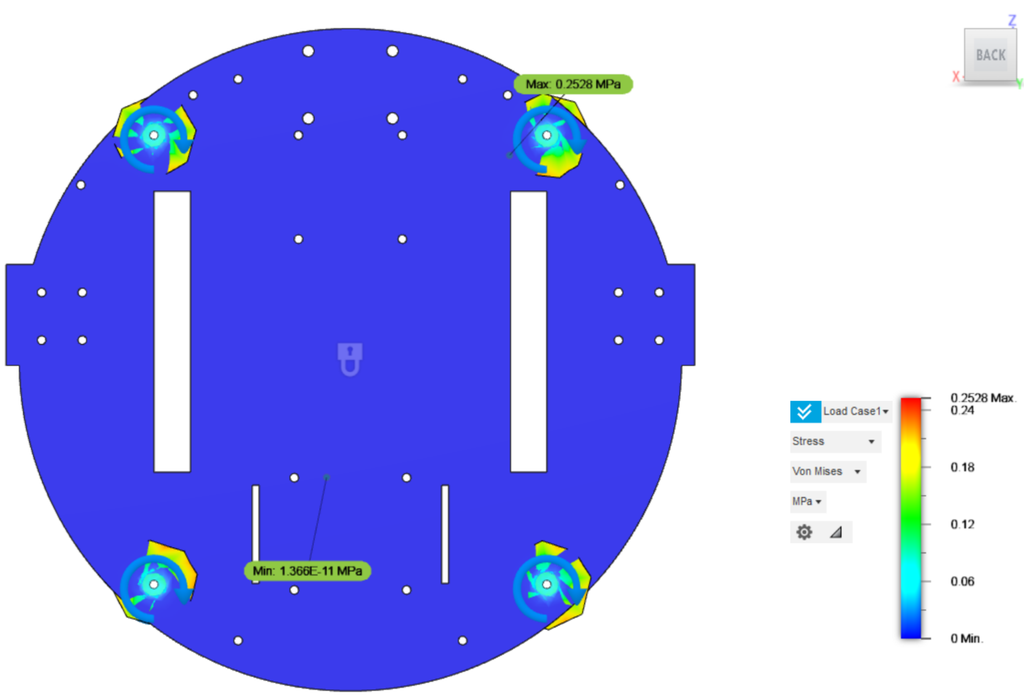

Figure 16: Simulation of Stress

And Acrylic's ultimate stress is 78.9Mpa. The maximum stress shown in 16 is 0.2528Mpa. This layer plate will not be broken. After we determine final positions of all components, we will do simulation first to see the weight distribution and balance.

3 Verification

3.1 Motion Unit

Requirement	Verification	Status
1. The robotic system should possess the capability to execute translational movements in both the forward and reverse directions, as well as rotational movements about its vertical axis to the left and right, while maintaining stability and balance. 2. The robotic system should be equipped with advanced sensor technology and intelligent control algorithms that enable it to autonomously detect and navigate around obstacles in its environment, without requiring human intervention.	1. We conducted a series of tests in a controlled environment, by adjusting the duty cycle, we enable the electronic pet to move forward and backward in a straight line on a flat road and to turn at a steady rate. 2. We conduct a series of obstacle avoidance tests in a controlled environment by placing obstacles of various sizes and shapes in the robot's path and observing its behavior as it approaches the obstacles. We recorded and analyzed the robot's movements and sensor data to make sure it is able to accurately detect the obstacles and navigate around them without colliding or requiring human intervention.	Pass

3.2 Voice Control Unit

Requirement	Verification	Status
1. The robotic system can activate non-plug-in function and realize AI conversation. 2. The robotic system can active plug-in function and enter immersive mode or non-immersive mode.	1. We can ask robot different questions without keyword we set before. After several seconds, the answer or response from ChatGPT 3.5 can be given. 2. We conduct the keyword to active plug-in fuction. If robotic system enter immersive mode, all actions can be controlled by voice. Such as playing music. If robotic system enter non-immersive mode, it can only do the specified thing, once completed, ends the program and needs to be triggered again, like photo taken, tennis follwing and so on.	Pass

3.3 Face Tracking Unit

Requirement	Verification	Status
1. Hardware components: CSI Camera: Make sure we have a compatible CSI camera module that can be connected to the Raspberry Pi. Pan-Tilt Mechanism: we need a pan-tilt mechanism to control the movement of the camera. This mechanism allows the camera to rotate horizontally (pan) and vertically (tilt) to track the detected face accurately. Ensure the pan-tilt mechanism can be controlled programmatically. 2. Software and libraries: Face Detection Algorithm: Choose a reliable face detection algorithm that can accurately detect faces in real-time. Popular options include deep learning-based models like OpenCV's DNN module with pre-trained models. Operating System and Software Environment: Install a suitable operating system like Raspbian on the Raspberry Pi. Set up the required software environment with Python and relevant libraries like OpenCV for image processing, camera access, and pan-tilt control libraries RPi.GPIO for controlling the pan-tilt mechanism.	1. Hardware verification: Verify that the camera can be connected to the Raspberry Pi correctly with $detect = 1$. Raspberry Pi Compatibility: Verify that the Raspberry Pi model you have selected meets the minimum requirements for running the face detection algorithm and controlling the pan-tilt mechanism. Pan-Tilt Mechanism Compatibility: Check the mechanical and electrical specifications of the pan-tilt mechanism and confirm that it can interface with the Raspberry Pi successfully. 2. Software verification: Face Detection Algorithm Performance: Test the selected face detection algorithm using sample images or a camera feed to verify its accuracy and real-time performance. Ensure that it can reliably detect faces under various lighting conditions and orientations. Software Environment Setup: Install the required software libraries and dependencies on the Raspberry Pi. Verify that the camera access, image processing, and pan-tilt control libraries are installed correctly and functioning as expected. Run sample programs or scripts to confirm their functionality.	Pass

4 Cost

The labor cost for our project is estimated as follows. The estimated salary is ¥60 per hour per person, following the salary standard for Zhejiang University undergraduates. We expect 20 hours of contribution from each team member each week, due to the complexity of our design. This project is expected to be completed in 16 weeks. With the assumptions given above, the calculation of labor cost is as follows.

$60/hr \times 2.5 \times 15hr/week \times 16week \times 4people = 144000$

Table 1: Cost Analysis of Components

Description	Mft	Part #	Qty	Cost	Total
3.7V 6000mAh lithium battery	Delipow	18650	2	26.9	45.4
4GB Raspberry Pi	Raspberry Ri Foundation	4B	1	998	998
2WD trolley with 3-6V motors	Jiaxingwei	JXINW	1	15.8	15.8
3.3/5V Pan-Tilt Hat	Shenzhen Continental Electronics	PCA9685 & TSL2581	1	97	97
32G 120mb/s tf card	SanDisk	ZN6MA	1	28.8	28.8
1A battery protect	TELESKY	534316461	1	3.59	3.59
500w pixels camera	Dalysheng	Raspberry Pi Camera	2	19.8	39.6
3.3-5V infrared sensor	Risym	E18D80NK	6	3.72	22.32
1.4-5.5V 2mA ultrasonic module	Risym	US-100	1	16.4	16.4
Motor drive module	Risym	L298N	1	9.8	9.8
Boost module	Fenglinwanshen	UPS-15W	2	16.5	33
1.15 inch OLED	Waveshare	SSD1309	1	104	104
Bluetooth speaker	MIUI	Portable Version	1	49	49
					1462.71

5 Conclusion

5.1 Accomplishments

components and functionalities to enhance its capabilities. The accomplished design includes the following features:

1. **Obstacle Avoidance:** The integration of three ultrasonic sensors and two infrared sensors enables the electronic pet to navigate its environment and avoid obstacles effectively. These sensors detect objects in the pet's path and provide feedback to the control system, allowing the pet to adjust its movements and path to prevent collisions.

2. **Face Tracking:** The inclusion of a pan-tilt hat and a CSI camera enables the electronic pet to track and follow detected faces. The pan-tilt hat allows the camera to rotate horizontally (pan) and vertically (tilt) to accurately track the detected face's movements. The CSI camera captures real-time images, which are processed using face detection algorithms to identify and track faces within the pet's field of view.

3. **Interaction and Engagement:** By combining face tracking capabilities with obstacle avoidance, the electronic pet can interact with users in a more dynamic and engaging manner. It can autonomously navigate towards detected faces while ensuring it avoids obstacles in its path, creating an interactive and immersive experience for users.

4. **Sensor Integration and Data Processing:** The design successfully integrates multiple sensors, including ultrasonic and infrared sensors, to provide comprehensive environmental perception. The data collected by these sensors is processed and analyzed by the control system to make informed decisions regarding obstacle avoidance and movement adjustments.

5. **Safety and Reliability:** The integration of obstacle avoidance sensors ensures the electronic pet operates safely and reliably in various environments. By detecting and avoiding obstacles, the pet reduces the risk of collisions and potential damage to itself or its surroundings, enhancing the overall safety and longevity of the device.

The accomplishment of this design demonstrates a sophisticated electronic pet with the ability to autonomously navigate its environment, avoid obstacles, and track and engage with detected faces. This combination of features creates an interactive and immersive experience for users while prioritizing safety, reliability, and user-friendly control.

5.2 Uncertainties

The design of the electronic pet, although functional, may encounter certain uncertainties and areas for improvement. The following uncertainties are associated with the design:

1. **Face Recognition Algorithm Speed:** The speed of the existing face recognition algo-

rithm may not be fast enough to ensure real-time and responsive tracking of faces. There is a possibility that the algorithm's processing time could delay the pet's ability to accurately track and follow detected faces, leading to suboptimal performance and decreased user experience.

2. **Sensor Sensitivity:** The ultrasonic and infrared sensors incorporated into the design may not be sensitive enough to reliably detect obstacles or environmental cues. This limitation could result in false readings, inaccurate obstacle avoidance, or insufficient responsiveness to the pet's surroundings. Improving the sensitivity of these sensors is crucial to enhance the pet's ability to detect and navigate obstacles effectively.

3. **Camera Data Transmission Speed:** The design may encounter limitations in the speed of camera data transmission. If the data transfer rate between the CSI camera and the processing unit is slow, it can cause delays in real-time image processing and face tracking. This can impact the pet's responsiveness and tracking accuracy.

4. **Motor Stability:** The operation of motors responsible for controlling the pan-tilt mechanism may suffer from instability. This instability can manifest as jittering, inaccuracies in movement, or inconsistent tracking of detected faces. Ensuring stable motor operation is essential for precise and smooth movement, which directly impacts the pet's ability to track faces accurately.

Addressing these uncertainties will be critical for further improving the design of the electronic pet. Possible solutions may include:

1. Exploring and implementing faster face recognition algorithms optimized for real-time performance to enhance the pet's responsiveness in tracking faces.

2. Conducting thorough sensor calibration and testing to improve the sensitivity and reliability of the ultrasonic and infrared sensors for accurate obstacle detection.

3. Investigating alternative camera data transmission methods or optimizing existing protocols to increase the speed of image processing and face tracking.

4. Conducting a comprehensive analysis of the motor control system to identify and address issues causing instability. This may involve fine-tuning motor parameters, using more robust motors, or implementing advanced control algorithms to ensure stable and precise movements.

By addressing these uncertainties and making necessary improvements, the design can overcome limitations and achieve enhanced performance, providing a more seamless and immersive experience for users.

5.3 Ethics

Data privacy: AI pets, such as robotic companions or virtual creatures, raise significant concerns regarding data privacy. The collection and storage of personal data, including images, audio recordings, and potentially sensitive information, necessitate secure handling to prevent unauthorized access or misuse. Safeguarding user privacy rights, implementing transparent data usage policies, and employing robust security measures are critical ethical considerations in the development of AI pets.

Accessibility: Ensuring the accessibility of AI pets to a diverse range of users, including individuals with disabilities, is an important ethical consideration. Incorporating features like audio or haptic feedback and supporting assistive technologies enhances inclusivity and usability, promoting equitable access to AI pet technologies.

Security: The security of AI pets against potential cyber threats, such as hacking or malware, is of utmost importance. Employing encryption, firewalls, and user authentication mechanisms helps protect user data and safeguards the integrity and functionality of AI pets, preventing unauthorized access or malicious activities.

Environment: Considering the environmental impact of AI pets is an ethical imperative. The operation of physical AI pets may consume energy and resources, leading to ecological consequences. Mitigating the ecological footprint of AI pets through energy-efficient design, sustainable manufacturing practices, and responsible resource usage aligns with environmental sustainability goals.

Interpersonal interaction: The potential influence of AI pets on human behavior and relationships presents ethical considerations. AI pets are designed to interact with humans, evoking emotional responses and shaping social dynamics. Ensuring these interactions promote positive, respectful, and healthy emotional connections without substituting genuine human relationships or causing isolation is vital. Ethical considerations involve understanding the impact on human social interactions and relationships to enhance overall well-being.

5.4 Future Work

1. **Advanced Face Detection Algorithms**: Explore and implement state-of-the-art face detection algorithms that offer improved accuracy and robustness in detecting faces under various conditions. Deep learning-based models or hybrid approaches combining multiple algorithms could be investigated to enhance the face detection capabilities of the system.

2. **Real-time Facial Recognition**: Integrate facial recognition capabilities into the system to not only detect faces but also identify specific individuals. This would require developing or utilizing advanced machine learning models and databases for accurate and efficient real-time facial recognition.

3. **Pose Estimation and Tracking**: Enhance the system to estimate and track facial poses and movements accurately. This could involve implementing algorithms for pose estimation and tracking, such as using facial landmarks or 3D modeling techniques. Precise tracking of facial movements would enable more dynamic and responsive interactions with the detected face.

4. **Multi-Face Tracking**: Extend the system to handle multiple faces simultaneously.

This would involve developing algorithms that can track and distinguish between multiple faces in real-time, enabling the unit to interact with and track multiple individuals within the camera's field of view.

5. **Integration of Emotion Recognition**: Integrate emotion recognition capabilities into the system to detect and interpret facial expressions. This could involve training machine learning models to recognize and classify different emotional states based on facial cues. Understanding the emotional state of the detected faces could enable the unit to respond and interact in a more empathetic and personalized manner.

6. **Optimized Pan-Tilt Mechanism**: Improve the hardware design of the pan-tilt mechanism to enhance its accuracy, speed, and smoothness of movement. This could involve using more precise motors, advanced control algorithms, or even exploring alternative mechanisms for better tracking performance.

7. **Intuitive User Interface**: Develop a user-friendly interface that allows users to easily configure and control the face tracking unit. This could involve designing a graphical interface or developing a mobile application that provides intuitive controls for adjusting tracking parameters, selecting tracking modes, and accessing additional functionalities.

8. **Power Efficiency and Portability**: Improve the power efficiency of the system to extend its battery life or reduce power consumption when connected to a power source. Additionally, consider optimizing the form factor and design of the unit to make it more portable and versatile, allowing for easy deployment and usage in various environments.

References

[1] W. Pan, *Wukong-robot*, GitHub, Jan. 2019. [Online]. Available: https://github.com/wzpan/wukong-robot (visited on 05/18/2023).

[2] , *Kitt-ai/snowboy*, GitHub, May 2021. [Online]. Available: https://github.com/Kitt-AI/snowboy (visited on 05/19/2023).

[3] *Baidu ai open platform*, ai.baidu.com. [Online]. Available: https://ai.baidu.com/.

[4] rany, *Edge-tts*, GitHub, May 2023. [Online]. Available: https://github.com/rany2/edge-tts (visited on 05/18/2023).